

Quantum knowledge for cryptographic reasoning

Vincent Danos¹ and Ellie D'Hondt²

In this paper we investigate security issues of quantum key distribution (QKD) [4] from a knowledge-based perspective. To our knowledge this type of research has not been conducted before. Because of space limits we refer to rather than repeat earlier results which we build upon in this paper. In particular we rely on distributed quantum networks [1], a formal framework for distributed quantum computations, to describe QKD, and on quantum knowledge as defined in [3]. We assume some familiarity with quantum computation and reasoning about knowledge – for the reader not acquainted with these domains, we refer to the excellent [8] and [6].

Quantum key distribution. The goal of quantum key distribution is to establish a shared secret key between Alice and Bob. It is a private key distribution protocol that is secure against eavesdroppers [7]. In the notation of [1], the full specification of one step of the protocol is given by

$$QKD = \mathbf{A}(a) : \{1\}.[(c!b)(c?a).M_1H_1^a] \mid \mathbf{B}(b) : \{2\}.[(c?b)(c!a).M_2H_2^b] \mid E_{12} . \quad (1)$$

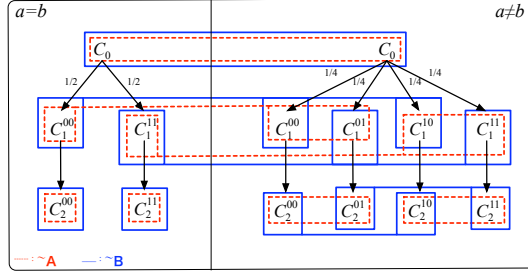
and is called the *QKD network*. The syntax for networks is a mix of classical process calculi syntax, the measurement calculus [2], and newly defined syntax for notions particular to the setting of distributed quantum computations. The measurement calculus is used for local quantum operations; we overload that notation slightly here by writing H_i^a , execution of the Hadamard gate conditioned by a . At the network level, we have two agents \mathbf{A} and \mathbf{B} acting in parallel on a shared quantum state given by E_{12} . At the agent level, agent \mathbf{A} (\mathbf{B}) is parameterised by a Boolean variable a (b) and owns qubit 1 (2), to which it applies the local quantum operation $M_1H_1^a$ ($M_2H_2^b$). After this \mathbf{A} and \mathbf{B} exchange the values of a and b by classical message passing. Alice and Bob only keep their measurement outcomes after checking that $a = b$. In this case it is guaranteed that Alice's measurement outcome s_1 equals that of Bob s_2 . These values are then kept as part of a secret key shared between Alice and Bob. A secret key of adequate length is established by iterating this protocol many times.

As the computation proceeds each agent's event sequence is updated along with its local state (storing measurement outcomes and classical variables). We need to specify the possible configurations of the network as it is this set we reason upon to analyse the knowledge of agents. Schematically we have two possible structures of configuration trees for QKD, two of each type. These are represented in the figure below (ignore the boxes for now) for $a = b$ or $a \neq b$ respectively. Here we have not written the dependency of each configuration on a and b explicitly to avoid cluttering the picture. Our configuration trees are slightly cruder than the ones taking all steps of the protocol into account because we allow local quantum operations to be carried out in parallel. Since

¹ Université Paris 7, France, Vincent.Danos@pps.jussieu.fr

² Vrije Universiteit Brussel, Belgium, Ellie.DHondt@vub.ac.be

the order in which local quantum operations are carried out does not matter, we can make this simplification.



Quantum knowledge. In order to define quantum knowledge, we need to define an equivalence relation on configurations for each of the agents, embodying what an agent knows about the global configuration from its own information only. We deliberately do not say *local* information here, as, via the network preparation, an agent may also have non-local information at its disposal, under the form of correlations. Each agent’s equivalence relation has to reflect what an agent knows about the network state, the execution of the protocol and the results of measurements. This is captured by an agent’s event sequence together with its local state in a particular configuration. Hence two configurations are equivalent for an agent if the event sequence and the local state for that agent are identical in both configurations. We then define what it means for an agent \mathbf{A}_i to know a fact θ in a configuration C in the usual way,

$$C, \mathcal{N} \models K_i \theta \iff \forall C' \sim_i C : C' \models \theta . \quad (2)$$

where \sim_i is agent \mathbf{A}_i ’s equivalence relation. The primitive propositions θ usually depend on the network under study, and are currently specified in an ad-hoc manner. One typically also wants to investigate how knowledge *evolves* during a computation. We use the approach of *computational tree logic* (CTL) [5] to formalise time-related logical statements, providing state as well as path modal operators. Concretely, we introduce the traditional temporal state operators \square (“always”) and \diamond (“eventually”) into our model, and combine these with the path operators A (“for all paths”) and E (“there exists a path”). For formal definitions of all the above we refer the reader to [3].

In the above figure the equivalence classes of configurations for Alice (red dotted boxes) and Bob (blue boxes) in the QKD network are pictured. Equivalence across both trees holds for Alice for a fixed, and for Bob for b fixed. Vertically, each level is discerned by each agent because a local event has occurred, while horizontally, different measurement outcomes ensure the non-equivalence of configurations. Note that at each time step *all* configurations in which say, Alice, has a particular measurement outcome are equivalent, i.e. this works across all configuration trees with the same value for a . More formally, we have the following two equivalence classes of configurations for Alice at level 1, one for each value of s_1 ,

$$[C_1^{s_1}]_{\mathbf{A}} = \{C_1^{s_1 s_2}(a, b), \forall b, s_2\} . \quad (3)$$

At the final level Alice and Bob have interchanged the values of a and b and so they can tell the difference between a number of configurations that were equivalent at the previous level. Concretely we move to a set of equivalence classes for Alice as follows,

$$[C_2^{s_1}(a, b)]_{\mathbf{A}} = \{C_2^{s_1 s_2}(a, b), \forall s_2\} . \quad (4)$$

Bob has analogous equivalence classes $[C_1^{s_2}(b)]_{\mathbf{B}}$ and $[C_2^{s_2}(a, b)]_{\mathbf{B}}$, which are defined as the above but with the roles of a and b and of s_1 and s_2 interchanged. We see that when $a = b$, Alice and Bob have identical single-configuration equivalence classes at the final stage of the computation. This means that each agent's knowledge is identical at that point, and that we can prove formally that in this case one bit of a secret key has been established,

$$a = b \Rightarrow C_0(a, b), QKD \models A \diamond (K_{\mathbf{A}}(s_1 = s_2) \wedge K_{\mathbf{B}}(s_1 = s_2)) \quad (5)$$

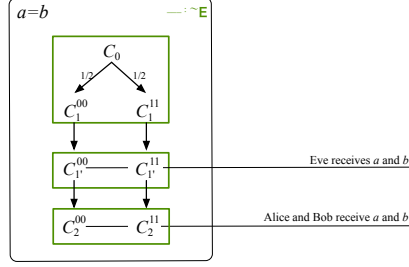
Security issues for QKD. We head off by showing that eavesdroppers listening in on classical communication channels cannot derive the key. This is of course the core property in making QKD so valuable as an alternative to classical private symmetric secret key establishment protocols. Suppose we have a third agent Eve who eavesdrops on classical communication between Alice and Bob. We can model this by adding a third agent $\mathbf{E}.$ to the QKD network given in (1). Note that Eve has no type since she has no qubits. The idea is that Eve intercepts the messages sent by Alice and Bob, reads them, and sends them further on, so that the other agents cannot detect her presence. However, neither can she derive any information about the key. This we know already from straightforward quantum mechanics, but let us prove this now in our framework. By including Eve an extra level is inserted in our configuration trees, between level 1 and 2. Calling this level 1', we thus obtain a number of new configurations $C_{1'}$ that our agents can reason upon. A representation of the $a = b$ case is given below. The crucial point is that, since the values of s_1 and s_2 remain local throughout the protocol, configurations with different values for these parameters are equivalent for Eve. In particular for $i = 1', 2$ we find the following equivalence classes

$$[C_i(a, b)]_{\mathbf{E}} = \{C_i^{s_1 s_2}(a, b), \forall s_1, s_2\}. \quad (6)$$

In other words, Eve can never derive the values of the secret key, since within one equivalence class there will always be configurations corresponding to different measurement outcomes. More formally,

$$a = b \Rightarrow C_0(a, b), QKD \models A \square \neg K_{\mathbf{E}}(s_1 = s_2 = v) \quad (7)$$

where v is a value. Note, however, that Eve can derive that in this case measurement outcomes for Alice and Bob are identical.



A second possible attack is that Eve intercepts the Bell state before it is distributed among Alice and Bob, measuring it and resending it to Alice and Bob in collapsed form. For simplicity, we assume that Eve always measures in the ordinary basis. This would involve changing the specification of the QKD network given in (1), as follows,

$$QKD = \mathbf{A}(a).[(c!b)(c?a).M_1 H_1^a(\text{qc?}1)] \mid \mathbf{B}(b).[(c?b)(c!a).M_2 H_2^b(\text{qc?}2)] \quad (8)$$

$$\mid \mathbf{E}.[(\text{qc!}1)(\text{qc!}2).M_{12}] \parallel E_{12} .$$

Note that Alice and Bob now have an empty type, as they receive their qubits at the start of the computation from Eve (whom they assume to be a safe quantum channel). The interesting case again is where $a = b$, and in particular $a = b = 1$ since Eve always uses the ordinary basis. In this case Alice and Bob may obtain different measurement outcomes even though they measure in the same basis. Hence they can figure out that an attack has been attempted, and thus that their network is not secure.

Again we have an extra level of configurations, this time right after $C_0(a, b)$, where Eve measures and sends qubits 1 and 2. However, let us move straight to the final stage of the computation. Now we have the following final equivalence classes for Alice

$$[C_2^{s_1}(1, 1)]_{\mathbf{A}} = \{C_2^{s_0 s_1 s_2}, \forall s_0, s_2\}, \quad (9)$$

where s_0 is Eve's measurement outcome. Indeed, since Eve's and Bob's measurement outcomes are now uncorrelated with Alice's, and they have not communicated these values to Alice, she has no way of discerning between configurations where her outcome s_1 is fixed but theirs are arbitrary. Thus the following statement holds:

$$C_0(1, 1), QKD \models A \diamond \neg K_{\mathbf{E}}(s_1 = s_2) . \quad (10)$$

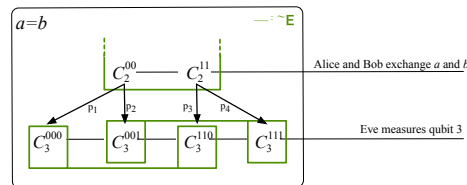
While in this case Alice and Bob cannot derive whether or not their protocol was safely executed, they can do so when they exchange their measurement outcomes s_1 and s_2 . In this case we have

$$C_0(1, 1), QKD \models E \diamond K_{\mathbf{E}}(s_1 \neq s_2) , \quad (11)$$

and hence they know without doubt that the security of their network was compromised.

A third possibility is that Eve entangles herself with the Bell state shared by Alice and Bob and operates on her qubit only. This is described by adding agent

$|E : \{3\}.O_3$ to the QKD network, where O is an arbitrary quantum operation, and changing the entanglement resource to $||\psi_{123}\rangle$, an arbitrary 3-qubit state which behaves like the Bell state on the first two qubits. Suppose O implements some unitary U acting on qubit 3 only. Since measurement statistics for Alice and Bob are unchanged, in the interesting case where $a = b$ we again have a situation as in the first attack, and hence (7) holds: Eve cannot derive the secret key. Next, suppose O implements a measurement of some sort. Only if qubit 3 is disentangled from the other two are the measurement statistics for Alice and Bob unaltered. Supposing Eve measures in the last step of the computation, then in this case with $a = b$, a final level of configurations is added to the configuration tree as pictured below. Since configurations with different values for the key are equivalent to Eve, again she cannot derive its value. Finally, if Eve carries out a measurement but her qubit is entangled with qubits 1 and 2, the measurement statistics are altered much in the way as when Eve measured the Bell state (only the probabilities are different). As we do not reason about probabilities, the analysis of that attack applies here as well.



References

1. Vincent Danos, Ellie D'Hondt, Elham Kashefi, and Prakash Panangaden. Distributed measurement-based quantum computation. In Peter Selinger, editor, *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005)*, volume 170 of *ENTCS*, pages 73–94, 2005. quant-ph/0506070.
2. Vincent Danos, Elham Kashefi, and Prakash Panangaden. The measurement calculus. *Journal of the ACM*, 2007. quant-ph/0704.1263v1.
3. Ellie D'Hondt and Prakash Panangaden. Reasoning about quantum knowledge. In *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 3821 of *LNCS*, page 0544c (to appear), 2005. quant-ph/0507176.
4. Arthur K. Ekert. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.*, 67(6):661–663, 1991.
5. E. Allen Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, pages 995–1072. MIT Press, 1990.
6. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about knowledge*. MIT Press, 1995.
7. Dominic Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, May 2001.
8. Michael A. Nielsen and Isaac Chuang. *Quantum computation and quantum information*. Cambridge university press, 2000.